

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Caching User Network Access
Information within a Network**

Inventor(s):

Murli D. Satagopan

Colin H. Brace

Mark R. Brown

ATTORNEY'S DOCKET NO. MS1-678US

1 **TECHNICAL FIELD**

2 This invention relates to a networked system and, in particular, to a
3 networked system in which network access information maintained at a central
4 location is distributed within the network.

5
6 **BACKGROUND**

7 In a network configuration, a global networked server can be implemented
8 to maintain a directory of all universal group memberships within the network for
9 each user authorized to access the network. A single directory of user network
10 access information maintained at a central location augments access security of the
11 network. An example of such a network configuration would be a company
12 having a headquarters site and one or more remotely located branch sites. The
13 server maintaining the directory of universal group memberships is implemented
14 at the headquarters site. Domain controllers are network servers that administrate
15 network access to clients and/or users at the remotely located branch sites.

16 The networked server is a global information server implemented as a
17 repository of global information for the network. A network can encompass many
18 domains where each domain is a unit of security. The global information server
19 maintains information about all of the domains in the network and provides one
20 central information store that can be queried by the domain controllers at the
21 networked branch sites to locate and access network-wide information and
22 resources.

23 A domain controller maintains information pertaining only to the domain or
24 domains that it is authoritative for. A domain administrator can designate users
25 and computers within a domain as security principals, and define groups of

1 security principals within a domain. A network administrator can define universal
2 groups having a membership of security principals that can be from many different
3 domains. Groups of security principals can be granted access to network
4 resources if the group memberships of a given user account are known.

5 A domain controller of a company branch office maintains user account
6 information pertaining to the users that access the company network at the
7 particular branch office. The complete set of universal group memberships for the
8 branch office user accounts, and for all domains in the network, however, are only
9 available at the global information server.

10 Each domain controller maintains a user object for each user authorized to
11 access the network from within a particular domain. In the example of the
12 company having remotely located branch sites, each branch site is distinguished as
13 a separate domain. However, two or more branch sites can be encompassed in,
14 and administrated as, a single domain.

15 A domain designates a replication partition and a security unit, and is not
16 bound by physical or geographic constraints. Typically, the size of a domain is
17 constrained by the number of users which represent a replication unit connected
18 through a low bandwidth link. For a low bandwidth link, it is preferable to
19 establish a small domain. Similarly, it would be disadvantageous to implement a
20 global information server at a location constrained by low bandwidth links.

21 The global information server maintains the directory of all universal group
22 memberships and replicates a copy of all the user objects from every domain
23 within the network. The server associates, in the directory, each replicated user
24 object with the universal group memberships that each user is authorized to access
25 in the network.

1 When a user attempts to logon to the network at a remotely located branch
2 site, the domain controller servicing the user's logon request at the particular
3 branch site validates the user name and password with an associated user object
4 maintained at the domain controller. The domain controller then evaluates the
5 user's universal group membership status prior to allowing the logon request. The
6 domain controller does so by sending a request to the global information server
7 where the directory having the universal group memberships that the user is a
8 member of is maintained.

9 If the global information server maintaining the directory is not available to
10 service the request from the domain controller, or if the communication link
11 between the domain controller and the server fails (is too slow, has an intermittent
12 connection, is unreliable, etc.), the user's logon request is denied. This is to
13 prevent a security breach of the network. Even though a user may have provided a
14 correct username and password, the logon request fails because the universal
15 group membership information is not available from the global information server
16 directory.

17 In such a network configuration, the universal group membership
18 information maintained in the global information server directory is required to be
19 available to each domain controller of the network to allow user logon and access
20 to the network. However, it is not practical and is cost prohibitive to implement a
21 local server to maintain a global group memberships directory at each branch
22 office site within the network due to limited hardware resources and available
23 network bandwidth constraints.

SUMMARY

A network system architecture has a global information server that maintains a directory of network access information that identifies users authorized to access the network system. The network system makes the network access information available to one or more domains of the network system.

A network domain controller at a branch site of the network system caches the network access information so that the domain controller can validate a network access request from a user without having to establish contact with the global information server. The domain controller tracks individual users that request access to the network system from the domain controller and refreshes the network access information for these users.

The domain controller refreshes the network access information for the users that have previously accessed the network system within a defined time interval. This prevents the cached network access information from becoming unreliable, and compromising network security. Refreshing the network access information for users that have previously accessed the network system from the domain controller within a defined time interval ensures that the latest available network access information is used at the domain controller to validate a network access request.

BRIEF DESCRIPTION OF THE DRAWINGS

The same numbers are used throughout the drawings to reference like features and components.

Fig. 1 is a block diagram of a network architecture.

1 Fig. 2 is a block diagram that illustrates a configuration of data structures in
2 the network architecture illustrated in Fig. 1.

3 Fig. 3 is a flow diagram of a method for caching user network access
4 information.

5 Fig. 4 is a flow diagram of a method for refreshing user network access
6 information

7 Fig. 5 is a block diagram that illustrates an alternative configuration of the
8 network architecture illustrated in Fig. 1.

9 Fig. 6 is a diagram of a computing system and environment that can be
10 utilized to implement the technology described herein.

11 12 **DETAILED DESCRIPTION**

13 The following technology describes systems and methods to provide user
14 network access information to one or more remote branch sites of a network. The
15 user network access information is periodically cached to a domain controller
16 from a network global information server that maintains the information. The user
17 access information is then available at a domain controller to validate a user
18 network access request irrespective of whether the global information server that
19 maintains the network access information is available to validate an access request
20 when initiated by a user. Furthermore, the systems and methods apply to any
21 types of information, resources, or data that is typically stored at a central location
22 within a network.

23 Fig. 1 shows a network architecture 100 in which a company has a
24 headquarters or main site 102 and two remotely located company branch sites 104,
25 106. Although the network architecture 100 is illustrated having only two branch

1 sites 104, 106, the methods and systems described herein are applicable to a
 2 network having one, or any number of, remotely located branch sites. The main
 3 site 102 and the branch sites 104, 106 are interconnected via a communications
 4 network 108. See the description of "Exemplary Computing System and
 5 Environment" below for specific examples of the network architectures and
 6 systems, computing systems, and system components described herein.

7 A network global information server 110 is implemented at the company
 8 main site 102. The server 110 maintains network-wide information and is
 9 communicatively linked to the company branch sites 104, 106 via the
 10 communications network 108.

11 The main site 102 and the branch sites 104, 106 each implement a network
 12 branch site domain controller to locally administrate network access and functions.
 13 Main site 102 has a domain controller 112 connected to the global information
 14 server 110. The main site 102 also has any number of work stations 114(1...x)
 15 connected to the domain controller 112. The work stations 114(1...x) facilitate
 16 user, client, or account access to the global information server 110 through the
 17 local domain controller 112. A global information server can also be implemented
 18 as a domain controller for one or more domains. Although the following
 19 description pertains mainly to user requests to access a network, it is to be
 20 appreciated that any type of account, user, user account, client, and the like can be
 21 part of a network architecture and request network access and network information
 22 and resources.

23 The branch site 104 has a domain controller 116 connected to the global
 24 information server 110 via the communications network 108. The branch site 104
 25 also has any number of work stations 118(1...y) connected to the domain

1 controller 116. The work stations 118(1...y) facilitate user access to the global
2 information server 110 through the local domain controller 116 (and via the
3 communications network 108). Similarly, branch site 106 has a domain controller
4 120 connected to the global information server 110 via the system network 108.
5 The branch site 106 has any number of work stations 122(1...z) connected to the
6 domain controller 120 to facilitate user access to the global information server 110
7 through the local domain controller 120 (and via the communications network
8 108).

9 Fig. 2 shows a configuration of data structures in a network architecture
10 200. The global information server 110 is connected to a network branch site
11 domain controller 202 via the communications network 108. A user work station
12 204 is locally connected (i.e., not via the communications network 108) to the
13 domain controller 202. The work station 204 supports a user interface 206 that
14 facilitates user access to the network 200 via the local domain controller 202.

15 The global information server 110 maintains a partial copy of every domain
16 in the network, where a domain is a replication partition boundary. The server 110
17 maintains universal group memberships for all of the domains in the network, and
18 potentially, the group memberships can be stored across multiple domains.

19 Domain controller 202 maintains a total copy of the one or more domains
20 for which it is authoritative. Global information server 110 maintains a directory
21 208 which is a copy of all the user objects 210(1...n) from every domain within
22 the network 200. The global information server directory 208 replicates a partial
23 copy of the user objects from every domain in the network and associates each
24 replicated user object 210(1...n) with the universal group memberships that each
25 user is authorized to access in the network 200. For example, domain controller

202 has a user object 1, identified as item 212, which is replicated in the directory 208 as the first user object 210(1) in the directory of user objects 210(1...n).

A data structure 214 illustrates the replicated information that is stored by the user objects 210(1...n) in the directory 208. The data structure 214 (i.e., a user object 210) has attributes 216 and metadata 218 associated with each attribute. An attribute User_Name 220 has user data 222 that associates a user of the network 200 with a user object 210. This attribute is replicated from the user objects maintained at the network domain controllers.

The data structure 214 also has an attribute Group_Memberships 224 and an attribute Site_Affinity 226. Group_Memberships 224 associates data that is a list of security identifiers (SIDs) 228 that denote the set of global and universal groups that a particular user object 210(1...n) is a member of. The Group_Memberships attribute 224 is not replicated from the user objects maintained at the network domain controllers. Rather, the SIDs 228 identify the groups that a particular user is a member of and are associated with each replicated user object 210(1...n) by the directory 208 to maintain network security. That is, the Group_Memberships 224 are user network access information that is centrally maintained by the global information server 110.

The Site_Affinity attribute 226 is multi-valued and associates data that is a data structure 230. Each Site_Affinity value has a globally unique identifier (GUID) 232 and a timestamp 234. Site_Affinity 226 conveys which networked branch site a particular user accesses the network 200 from. That is, the GUID 232 uniquely identifies the branch site that a user accesses the network 200 from and the timestamp 234 identifies the time at which the user requests access to the

1 network 200. This attribute is replicated from the user objects maintained at the
2 network domain controllers.

3 The domain controller 202, and each domain controller in a network,
4 maintains a user object for each user authorized to access the network from a
5 particular branch site. That is, for each user that accesses the network from a
6 workstation connected to the domain controller 202, such as workstation 204,
7 domain controller 202 maintains an associated user object for each user.

8 Domain controller 202 maintains the user object 212 that stores user
9 network access information for a user that requests access to the network 200 via
10 the domain controller. Similarly to the user object data structure 214 maintained
11 by the network directory 208, user object 212 at domain controller 202 has
12 attributes 236 and metadata 238 associated with each attribute. An attribute
13 User_Name 240 has user data 242 that associates a user that requests access to the
14 network 200 from a workstation connected to the domain controller 202. The
15 User_Name attribute 240 is replicated as User_Name 220 which is maintained in
16 the global information server directory 208 for each associated user object
17 210(1...n).

18 The user object 212 also has a Site_Affinity attribute 246 that associates a
19 multi-valued data structure 248. The Site_Affinity attribute 246, and the
20 multi-valued data structure 248, are the replicated attribute 226 and data structure
21 230 maintained in the global information server directory 208. That is, the multi-
22 valued data structure 230 is replicated from the data structure 248 maintained at
23 domain controller 202.

24 The user object 212 has a Cached_Membership attribute 250 that associates
25 SIDs 252 from the list of SIDs 228 maintained in the global information server

1 directory 208 for each particular user object 210(1...n). The domain controller
2 202 periodically caches the SIDs 228 from the global information server directory
3 208 and stores the user network access information at the user object 212 in the
4 Cached_Membership attribute 250.

5 The user object 212 also has a Last_Refresh_Time attribute 254 that
6 denotes an update time 256 which indicates when a given user's cached
7 membership information (i.e., the Cached_Membership SIDs 252) was last
8 updated, or refreshed. A periodic refresh of the user membership information is
9 needed to guarantee an upper bound on how old the membership information is.
10 For example, if a user does not request network access for an extended preset
11 period of time, the user's membership information may be unreliable, or not the
12 latest information available at the global information server 110. This presents the
13 possibility of compromising network security because the user may no longer be
14 authorized to access the network, yet the Cached_Membership SIDs 252 identify
15 that the user can still access the network.

16 A network domain controller has a set of registry keys that can be set to
17 control certain aspects of caching the user network access information. For
18 example, domain controller 202 has a registry 258 containing registry keys that
19 have default values, or the values can be changed, to control caching the SIDs 252
20 at the user object 212 from the global information server directory 208.

21 A Half_Life parameter 260 indicates one-half the maximum time for which
22 a particular user's membership information (i.e., the Cached_Membership SIDs
23 252) will be automatically refreshed without having a logon network access
24 request serviced by the domain controller 202. The network domain controller
25 202 maintains a user refresh list 262 of users whose membership information is

periodically refreshed based on the last time that a user requested access to the network 200. A user is deleted from the user refresh list 262 if the user has not requested access to the network 200 via a domain controller for a period of time that equals $2 \times \text{Half_Life}$. For example, the default value of the Half_Life interval can be set to three months. Thus, a particular user's membership information would not be refreshed if the user has not requested network access within a six month time period.

A Staleness parameter 264 indicates a maximum time after which the cached user group membership information (i.e., the Cached_Membership SIDs 252) will be considered "stale", or too old to be considered reliable with respect to network security. If the default value is one week, a user network access request will be failed if the cached membership information is older than this time period (and the global information server directory 208 is not available to service the network access request from a domain controller).

A Refresh_Interval parameter 266 indicates how frequently to update or refresh the Cached_Membership SIDs 252 from the global information server directory 208. For example, the default value to refresh the user group membership information can be set to eight hours. A Refresh_Limit parameter 268 controls the maximum number of users refreshed in every Refresh_Interval, which can be five-hundred users.

Fig. 3 illustrates a method for caching user network access information at a remotely located branch site domain controller and refers to items described in Figs. 1 and 2 by reference number. The order in which the method is described is not intended to be construed as a limitation. At block 300, a user requests access to a network 200 from a work station 204 connected to a network domain

1 controller 202. At block 302, the domain controller 202 validates the username
2 and password with user object 212 maintained at the domain controller. The user
3 object 212 is associated with the user requesting access to the network 200. If the
4 username and password supplied by the user are not validated (i.e., “no” from
5 block 302), the logon network access request is denied at block 304.

6 If the username and password supplied by the user are validated (i.e., “yes”
7 from block 302), domain controller 202 validates the user’s universal group
8 membership status. The domain controller does so by checking the user object
9 attribute Cached_Membership 250 for cached SIDs (security identifiers) 252 at
10 block 306.

11 If the associated user object 212 does have cached SIDs 252 (i.e., “yes”
12 from block 306), the domain controller 202 verifies that the Last_Refresh_Time
13 254 (i.e., update time 256) does not exceed the Staleness parameter 264 at block
14 308. If the cached SIDs 252 are still reliable (i.e., “yes” from block 308), the
15 domain controller 202 authorizes the user’s network access request with the
16 cached SIDs 252 at block 310.

17 If the user object 212 does not have cached SIDs 252 for the user
18 requesting network access (i.e., “no” from block 306), or if the
19 Last_Refresh_Time 254 exceeds the Staleness parameter 264 (i.e., “no” from
20 block 308), the domain controller 202 attempts to contact the global information
21 server 110 via the communications network 108 at block 312. If the domain
22 controller 202 cannot establish a communications link with the global information
23 server 110 (i.e., “no” from block 312), the logon network access request is denied
24 at block 314. If the domain controller 202 can establish a communications link
25 with the global information server 110 (i.e., “yes” from block 312), the domain

controller 202 authorizes the user's network access request with the SIDs 228 stored in the associated user object data structure 214 at the global information server 110 (block 316).

After contacting the global information server 110 to service the user's network access request (blocks 312, 316), the domain controller 202 updates the user object 212 with the user network access information maintained at the global information server directory 208 in the data structure 214. At block 318, the domain controller 202 updates Cached_Membership 250 by caching the SIDs 252 from the SIDs 228 stored in the global information server directory 208. The domain controller 202 also updates the Last_Refresh_Time 254 in user object 212 at block 320.

At block 322, the domain controller 202 updates the Site_Affinity attribute 246 (i.e., the multi-valued data structure 248) in user object 212. This indicates that a user access request is initiated at a particular branch site domain controller. The domain controller 202 updates the Site_Affinity attribute 246 if the multi-valued data structure 248 does not indicate that the networked branch site domain controller 202 is where the user requested network access. The Site_Affinity attribute 246 is also updated by the domain controller 202 if a user performs a password change operation on a user account associated with a user object at domain controller 202.

If the domain controller 202 cannot establish a communications link with the global information server 110 (i.e., "no" from block 312), and the logon network access request is denied at block 314, blocks 318-320 are performed as described above when the domain controller 202 can next establish a communications link with the global information server 110. This is to facilitate a

1 subsequent user access request if the user has tried to logon but failed. The
2 network access information for the user is updated for the next user access request.

3 Fig. 4 illustrates a method to periodically refresh user network access
4 information at a remotely located branch site domain controller and refers to items
5 described in Figs. 1 and 2 by reference numbers. The order in which the method is
6 described is not intended to be construed as a limitation. The domain controller
7 202 periodically refreshes the Cached_Membership SIDs 252 for users associated
8 with the domain controller. At block 400, the domain controller 202 identifies
9 those user objects (i.e., users or clients) at the domain controller having an affinity
10 for requesting network access via the domain controller. User objects are
11 identified by the Site_Affinity 246 GUID at the domain controller.

12 If a user is identified as having requested network access at domain
13 controller 202 (i.e., “yes” from block 400), the domain controller checks that the
14 associated Site_Affinity 246 Timestamp does not exceed $2 \times \text{Half_Life}$ parameter
15 260 at block 402. If a user is not identified as requesting network access a domain
16 controller 202 (i.e., “no” from block 400), or if a user has not requested network
17 access within a set period of time (i.e., “no” from block 402), another user object
18 maintained at the domain controller 202 is evaluated at block 404 to evaluate if the
19 associated user network access information will be refreshed.

20 If a user has requested network access within a set time period (i.e., “yes”
21 from block 402), the domain controller 202 updates the user refresh list 262 at
22 block 406. The user objects in the refresh list indicate those users having a site
23 affinity for the domain controller 202, and a Timestamp that has not expired.

24 At block 408, the domain controller 202 determines if all of the user objects
25 at the domain controller have been evaluated to determine if the associated user

1 network access information will be refreshed. If the domain controller 202 has not
2 evaluated all of the user objects (i.e., "no" from block 408), another user object
3 maintained at the domain controller 202 is evaluated at block 404 to determine if
4 the associated user network access information will be refreshed.

5 If all of the user objects have been evaluated (i.e., "yes" from block 408),
6 the domain controller 202 establishes a communication link with the global
7 information server 110 via the communications network 108 at block 410. In a
8 network architecture having more than one global information server that
9 maintains a directory of user network access information, a domain controller can
10 refresh the user access information from whichever global information server is
11 available and/or is the most efficient connection by virtue of bandwidth and/or
12 cost. At block 412, the domain controller 202 updates the Cached_Membership
13 SIDs 252 and the Last_Refresh_Time 254 (update time 256) for each user object
14 identified to be updated in the user refresh list 262.

15 Fig. 5 illustrates an alternative configuration of network architecture 100
16 described in Fig. 1. Fig. 5 shows a network architecture 500 in which a company
17 has a main site 502 and two remotely located company branch sites 504, 506. The
18 main site 502 and the branch sites 504, 506 are interconnected via a
19 communications network 508. The network architecture 500 implements two
20 global information servers. A first global information server 510 is implemented
21 at the main site 502 and a second global information server 512 is implemented at
22 the branch site 506. Each server 510, 512 maintains network-wide information
23 and is communicatively linked within the network 500 via the communications
24 network 508. Although the network architecture 500 is illustrated having only two
25 global information servers 510, 512, the methods and systems described herein are

1 applicable to a network architecture having one, or any number of, global
2 information servers.

3 The main site 502 and the branch site 506 each implement a network
4 branch site domain controller to locally administrate network access and functions.
5 Main site 502 has a domain controller 514 connected to the global information
6 server 510. The main site 502 also has any number of work stations 516(1...x)
7 connected to the domain controller 514.

8 The branch site 506 has a domain controller 518 locally connected to the
9 global information server 512. The branch site 506 also has any number of work
10 stations (not shown) connected to the domain controller 518. The branch site 504
11 has two domain controllers 520, 522 interconnected locally at the branch site 504
12 and connected to each of the global information servers 510, 512 via the
13 communications network 508. The branch site 504 has any number of work
14 stations 524(1...y) connected to either or both of the domain controllers 520, 522
15 to facilitate user access to either of the global information servers 510, 512.

16 Both of the global information servers 510, 512, and each of the four
17 domain controllers 514, 518, 520, 522 function to cache and refresh user network
18 access information as described in reference to the configuration of data structures
19 in a network architecture as shown in Fig. 2, and as described in reference to the
20 methods shown in Figs. 3 and 4.

21 **Exemplary Computing System and Environment**

22 Fig. 6 illustrates an example of a computing environment 600 within which
23 the computer and network architectures described herein can be either fully or
24 partially implemented. Exemplary computing environment 600 is only one
25 example of a computing system and is not intended to suggest any limitation as to

1 the scope of use or functionality of the network architectures. Neither should the
2 computing environment 600 be interpreted as having any dependency or
3 requirement relating to any one or combination of components illustrated in the
4 exemplary computing environment 600.

5 The computer and network architectures can be implemented with
6 numerous other general purpose or special purpose computing system
7 environments or configurations. Examples of well known computing systems,
8 environments, and/or configurations that may be suitable for use include, but are
9 not limited to, personal computers, server computers, thin clients, thick clients,
10 hand-held or laptop devices, multiprocessor systems, microprocessor-based
11 systems, set top boxes, programmable consumer electronics, network PCs,
12 minicomputers, mainframe computers, distributed computing environments that
13 include any of the above systems or devices, and the like.

14 Caching user network access information may be described in the general
15 context of computer-executable instructions, such as program modules, being
16 executed by a computer. Generally, program modules include routines, programs,
17 objects, components, data structures, etc. that perform particular tasks or
18 implement particular abstract data types. Caching network access information
19 may also be practiced in distributed computing environments where tasks are
20 performed by remote processing devices that are linked through a communications
21 network. In a distributed computing environment, program modules may be
22 located in both local and remote computer storage media including memory
23 storage devices.

24 The computing environment 600 includes a general-purpose computing
25 system in the form of a computer 602. The components of computer 602 can

1 include, by are not limited to, one or more processors or processing units 604, a
2 system memory 606, and a system bus 608 that couples various system
3 components including the processor 604 to the system memory 606.

4 The system bus 608 represents one or more of any of several types of bus
5 structures, including a memory bus or memory controller, a peripheral bus, an
6 accelerated graphics port, and a processor or local bus using any of a variety of
7 bus architectures. By way of example, such architectures can include an Industry
8 Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an
9 Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA)
10 local bus, and a Peripheral Component Interconnects (PCI) bus also known as a
11 Mezzanine bus.

12 Computer system 602 typically includes a variety of computer readable
13 media. Such media can be any available media that is accessible by computer 602
14 and includes both volatile and non-volatile media, removable and non-removable
15 media. The system memory 606 includes computer readable media in the form of
16 volatile memory, such as random access memory (RAM) 610, and/or non-volatile
17 memory, such as read only memory (ROM) 612. A basic input/output system
18 (BIOS) 614, containing the basic routines that help to transfer information
19 between elements within computer 602, such as during start-up, is stored in ROM
20 612. RAM 610 typically contains data and/or program modules that are
21 immediately accessible to and/or presently operated on by the processing unit 604.

22 Computer 602 can also include other removable/non-removable,
23 volatile/non-volatile computer storage media. By way of example, Fig. 6
24 illustrates a hard disk drive 616 for reading from and writing to a non-removable,
25 non-volatile magnetic media (not shown), a magnetic disk drive 618 for reading

1 from and writing to a removable, non-volatile magnetic disk 620 (e.g., a “floppy
2 disk”), and an optical disk drive 622 for reading from and/or writing to a
3 removable, non-volatile optical disk 624 such as a CD-ROM, DVD-ROM, or other
4 optical media. The hard disk drive 616, magnetic disk drive 618, and optical disk
5 drive 622 are each connected to the system bus 608 by one or more data media
6 interfaces 626. Alternatively, the hard disk drive 616, magnetic disk drive 618,
7 and optical disk drive 622 can be connected to the system bus 608 by a SCSI
8 interface (not shown).

9 The disk drives and their associated computer-readable media provide non-
10 volatile storage of computer readable instructions, data structures, program
11 modules, and other data for computer 602. Although the example illustrates a hard
12 disk 616, a removable magnetic disk 620, and a removable optical disk 624, it is to
13 be appreciated that other types of computer readable media which can store data
14 that is accessible by a computer, such as magnetic cassettes or other magnetic
15 storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or
16 other optical storage, random access memories (RAM), read only memories
17 (ROM), electrically erasable programmable read-only memory (EEPROM), and
18 the like, can also be utilized to implement the exemplary computing system and
19 environment.

20 Any number of program modules can be stored on the hard disk 616,
21 magnetic disk 620, optical disk 624, ROM 612, and/or RAM 610, including by
22 way of example, an operating system 626, one or more application programs 628,
23 other program modules 630, and program data 632. Each of such operating
24 system 626, one or more application programs 628, other program modules 630,
25

1 and program data 632 (or some combination thereof) may include an embodiment
2 of a caching scheme for user network access information.

3 Computer system 602 can include a variety of computer readable media
4 identified as communication media. Communication media typically embodies
5 computer readable instructions, data structures, program modules, or other data in
6 a modulated data signal such as a carrier wave or other transport mechanism and
7 includes any information delivery media. The term “modulated data signal”
8 means a signal that has one or more of its characteristics set or changed in such a
9 manner as to encode information in the signal. By way of example, and not
10 limitation, communication media includes wired media such as a wired network or
11 direct-wired connection, and wireless media such as acoustic, RF, infrared, and
12 other wireless media. Combinations of any of the above are also included within
13 the scope of computer readable media.

14 A user can enter commands and information into computer system 602 via
15 input devices such as a keyboard 634 and a pointing device 636 (e.g., a “mouse”).
16 Other input devices 638 (not shown specifically) may include a microphone,
17 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and
18 other input devices are connected to the processing unit 604 via input/output
19 interfaces 640 that are coupled to the system bus 608, but may be connected by
20 other interface and bus structures, such as a parallel port, game port, or a universal
21 serial bus (USB).

22 A monitor 642 or other type of display device can also be connected to the
23 system bus 608 via an interface, such as a video adapter 644. In addition to the
24 monitor 642, other output peripheral devices can include components such as
25

1 speakers (not shown) and a printer 646 which can be connected to computer 602
2 via the input/output interfaces 640.

3 Computer 602 can operate in a networked environment using logical
4 connections to one or more remote computers, such as a remote computing device
5 648. By way of example, the remote computing device 648 can be a personal
6 computer, portable computer, a server, a router, a network computer, a peer device
7 or other common network node, and the like. The remote computing device 648 is
8 illustrated as a portable computer that can include many or all of the elements and
9 features described herein relative to computer system 602.

10 Logical connections between computer 602 and the remote computer 648
11 are depicted as a local area network (LAN) 650 and a general wide area network
12 (WAN) 652. Such networking environments are commonplace in offices,
13 enterprise-wide computer networks, intranets, and the Internet. When
14 implemented in a LAN networking environment, the computer 602 is connected to
15 a local network 650 via a network interface or adapter 654. When implemented in
16 a WAN networking environment, the computer 602 typically includes a modem
17 656 or other means for establishing communications over the wide network 652.
18 The modem 656, which can be internal or external to computer 602, can be
19 connected to the system bus 608 via the input/output interfaces 640 or other
20 appropriate mechanisms. It is to be appreciated that the illustrated network
21 connections are exemplary and that other means of establishing communication
22 link(s) between the computers 602 and 648 can be employed.

23 In a networked environment, such as that illustrated with computing
24 environment 600, program modules depicted relative to the computer 602, or
25 portions thereof, may be stored in a remote memory storage device. By way of

1 example, remote application programs 658 reside on a memory device of remote
2 computer 648. For purposes of illustration, application programs and other
3 executable program components, such as the operating system, are illustrated
4 herein as discrete blocks, although it is recognized that such programs and
5 components reside at various times in different storage components of the
6 computer system 602, and are executed by the data processor(s) of the computer.

7 **Conclusion**

8 Although the systems and methods have been described in language
9 specific to structural features and/or methodological steps, it is to be understood
10 that the technology defined in the appended claims is not necessarily limited to the
11 specific features or steps described. Rather, the specific features and steps are
12 disclosed as preferred forms of implementing the claimed invention.
13
14
15
16
17
18
19
20
21
22
23
24
25